Command and Control (C&C/C2)

Course Introduction

Alexis Ahmed

Senior Penetration Tester @HackerSploit
Offensive Security Instructor @INE

Course Topic Overview

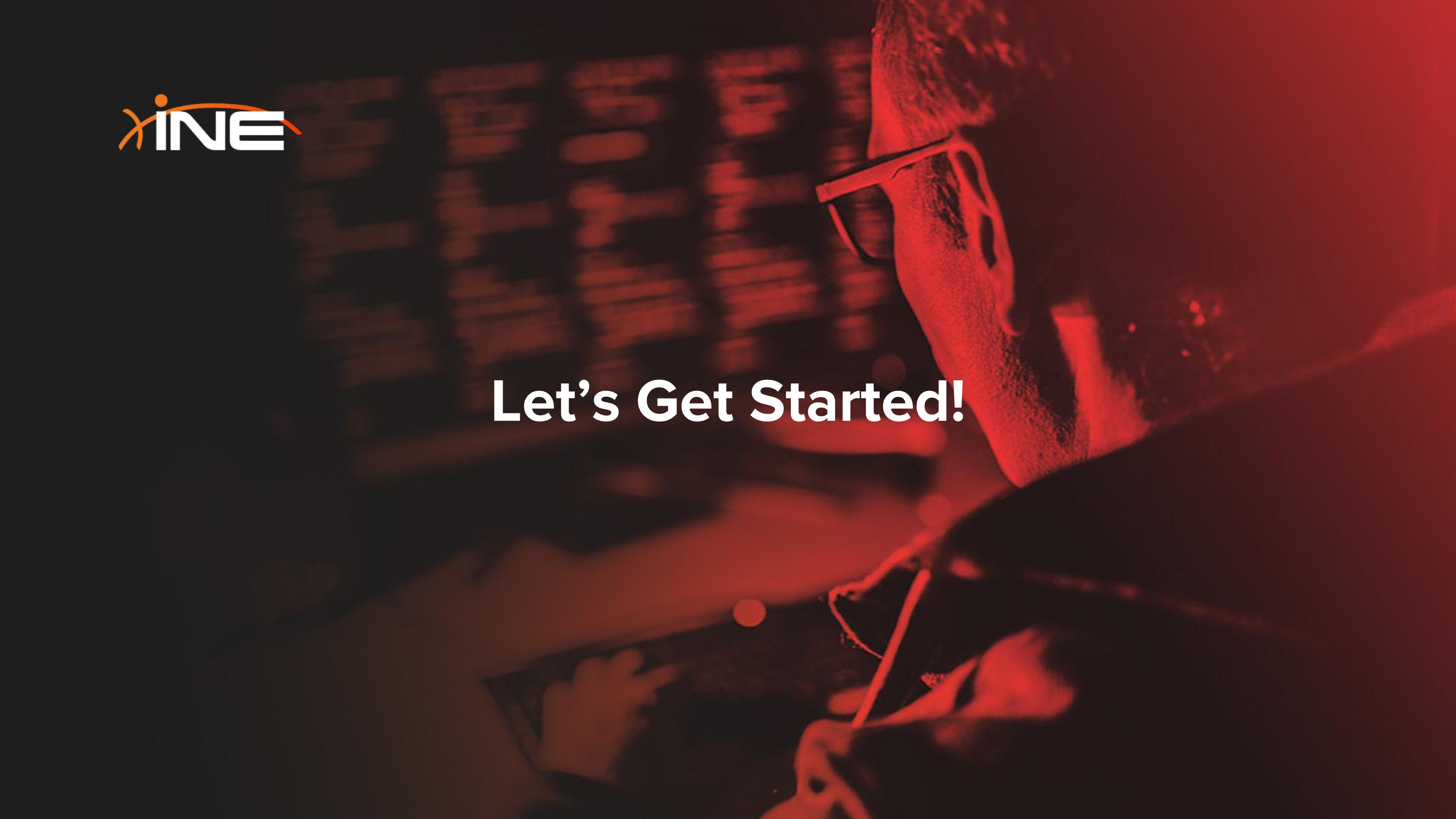
- Introduction to Command and Control (C&C/C2)
- How C2 Frameworks Work
- C2 Essential Terminology
- Deploying and Operating C2 Infrastructure
- Selecting the Correct C2 Framework
- Red Team Operations with PowerShell-Empire and Starkiller
- Exploring Other Popular C2 Frameworks

Prerequisites

- Knowledge and experience in penetration testing
- Familiarity with Windows and Linux
- Basic familiarity with the Metasploit Framework

Learning Objectives:

- You will have an understanding of what Command and Control is with regards to Red Team/Offensive operations.
- You will have an understanding of what C2 Frameworks are, how they work, what functionality they offer and the role they play in red team operations.
- You will have an understanding of the various communication models and protocols used in designing, deploying and operating C2 infrastructure .
- You will be able to identify the correct C2 Framework to use based on the nature of engagement you are performing in addition to the features you require for a successful operation.
- You will have the ability to install, configure and effectively use PowerShell-Empire and Stakiller for Red Team operations in Windows environments.
- You will have the knowledge and experience in using some of the most popular C2 Frameworks available.



Let’s Get Started!

Introduction to Command and Control

Command and Control

- Command and Control (C2 or C&C) refers to the communication structure used by attackers to remotely control and coordinate activities across compromised systems.

- It encompasses the methods, protocols, and infrastructure that enable attackers to send commands, receive data, and manage their operations in a coordinated manner.

- C2 typically consists of a central C2 server and client software (agents) installed on compromised endpoints.

Command and Control

- In red teaming, Command and Control (C2) plays a central role in simulating sophisticated cyberattacks.

- C2 frameworks allow red teams to orchestrate, control, and coordinate multi-stage offensive operations, providing them with the tools to mimic real-world threat actors.

Role of C2 in Red Teaming

Centralized Coordination:

- C2 frameworks provide a centralized point from which red teams can manage multiple compromised systems. This coordination is essential for controlling the flow of operations, issuing commands, and maintaining communication across distributed networks.

Persistence and Remote Access:

- C2 allows red teams to establish and maintain persistence on compromised systems. It enables remote access, allowing red teams to interact with target systems, execute commands, and carry out post-exploitation activities.

Role of C2 in Red Teaming

Lateral Movement:

- C2 frameworks facilitate lateral movement across a network. Red teams use C2 to compromise additional systems, navigate through network segments, and escalate privileges as they would in a real-world attack scenario.

Complex Attack Scenarios:

- C2 frameworks empower red teams to execute complex attack scenarios that span multiple stages. This includes tasks like data exfiltration, privilege escalation, and evasion techniques. By using C2, red teams can create realistic attack simulations.

Role of C2 in Red Teaming

Testing Blue Team Detection and Response:

- C2 frameworks are instrumental in testing an organization's defensive capabilities. Red teams can use C2 to challenge the blue team (defensive team) with realistic attacks, allowing them to evaluate detection and response effectiveness.

Importance of C2 in Red Teaming

- Realistic Simulation: C2 frameworks enable red teams to simulate advanced attack techniques that closely resemble those used by real-world threat actors. This realism is crucial for identifying gaps in an organization's security posture.

- Comprehensive Security Assessment: By leveraging C2, red teams can conduct a thorough assessment of an organization's defenses. This includes testing network segmentation, endpoint security, data loss prevention (DLP), and intrusion detection systems (IDS).

Importance of C2 in Red Teaming

- Identifying Weak Points: C2 frameworks allow red teams to explore a variety of tactics, techniques, and procedures (TTPs). This helps identify weak points in an organization's security infrastructure, providing valuable feedback for strengthening defenses.

- Improving Incident Response: Through red teaming exercises involving C2, organizations can test their incident response capabilities. This helps blue teams practice and refine their response processes, leading to faster and more effective reactions to actual threats.

Components of C2

- C2 Server: A central point where attackers issue commands and process data sent by the compromised systems. It can be hosted on various platforms, like on-premises servers, cloud services, or hidden services.

- C2 Agent: Software running on the compromised systems that connects to the C2 server. It facilitates remote control and data transmission.

- Communication Channel: The protocol and method through which the C2 server communicates with agents. This can include HTTP(S), DNS, WebSockets, custom encrypted protocols, etc.

Introduction To C2 Frameworks

C2 Frameworks

- A Command and Control (C2 or C&C) framework is a software platform designed for managing, controlling, and orchestrating the activities of remote systems or devices in an offensive security context. In cybersecurity,

- C2 frameworks are primarily used by attackers to maintain remote access to compromised systems, issue commands, and coordinate malicious activities. In ethical scenarios like penetration testing and red teaming, these frameworks enable security professionals to simulate advanced attack scenarios to assess an organization's security posture.

C2 Framework Functionality

Establishing Communication Channels

- C2 frameworks establish communication channels between the C2 server and compromised systems (agents/clients). These channels can use various protocols, such as HTTP, HTTPS, DNS, WebSockets, or custom protocols, often employing encryption and obfuscation to avoid detection.

Remote Control and Command Execution

- The primary function of a C2 framework is to enable remote control of compromised systems. It allows operators to execute commands on remote endpoints, access files, run scripts, and perform a variety of actions, mimicking the behavior of attackers.

C2 Framework Functionality

Persistence Mechanisms

- C2 frameworks offer features for maintaining persistence on compromised systems. This includes methods for ensuring that C2 agents persist through system reboots or other disruptions. Common persistence techniques involve modifying startup settings, creating scheduled tasks, or installing backdoors.

Lateral Movement

- A key capability of C2 frameworks is facilitating lateral movement across networks. They provide tools for compromising additional systems, exploiting network vulnerabilities, and navigating through different network segments.

C2 Framework Functionality

Privilege Escalation

- C2 frameworks often include modules or techniques for escalating privileges on compromised systems. This allows operators to gain higher-level access, enabling deeper control and broader impact on the target environment.

Data Exfiltration

- C2 frameworks can be used to collect and exfiltrate data from compromised systems. This is often achieved through specific commands or scripts that retrieve sensitive information and send it back to the C2 server.

C2 Framework Functionality

Automation & Scripting

- Many C2 frameworks support automation and scripting, allowing operators to create custom scripts and automate repetitive tasks. This capability is useful for streamlining complex operations and conducting coordinated attacks.

Evasion Techniques

- C2 frameworks offer evasion techniques to help operators avoid detection by security tools like firewalls, intrusion detection systems (IDS), and endpoint security solutions. This might involve obfuscating network traffic, using common ports, or employing domain fronting.

C2 Framework Functionality

Payload Development

- C2 frameworks often allow for custom payload development, giving operators flexibility to create unique payloads or modify existing ones. This enables red teamers to adapt their approach to specific scenarios or environments.

Logging & Reporting

- Many C2 frameworks provide logging and reporting capabilities, allowing operators to track commands, collect data on operations, and generate reports for analysis. This is useful for reviewing attack scenarios and sharing insights with stakeholders.

How C2 Frameworks Work

How C2 Frameworks Work

- Command and Control (C2) frameworks are designed to establish and maintain communication between a central command point (the C2 server) and distributed endpoints (the C2 agents or clients).

- This communication allows for issuing commands, transmitting data, and orchestrating various offensive activities. Here's an explanation of how C2 frameworks work, focusing on communication channels, protocols, and common communication models.

C2 Communication Channels

- Communication channels in a C2 framework enable data transmission between the C2 server and agents. These channels are crucial for remote control, task execution, and data exfiltration.

- C2 frameworks can use various communication channels, including:
  - Network-based Channels: Utilize network protocols like HTTP, HTTPS, DNS, FTP, WebSockets, or custom protocols to communicate between the server and agents.
  - Non-Network Channels: In rare cases, communication may occur via non-network methods like removable storage devices, though this is less common in penetration testing contexts.

C2 Communication Protocols

- Common Protocols: C2 frameworks frequently use standard network protocols to blend into regular network traffic. Commonly used protocols include:
  - HTTP/HTTPS: These are widely used because they are less likely to raise suspicion due to their ubiquity in web traffic. HTTPS adds encryption, providing a layer of security against interception.
  - DNS: This protocol is sometimes used for C2 communication due to its pervasiveness in networks, allowing data to be embedded in DNS queries or responses.
  - Custom Protocols: Some C2 frameworks use custom-designed protocols for enhanced flexibility or obfuscation.

C2 Communication Models

- C2 frameworks can follow different communication models to facilitate their operations.

- Here are some common models:
  - Centralized Model
  - P2P
  - Out of Band/Overt
  - Hybrid

Centralized C2 Model

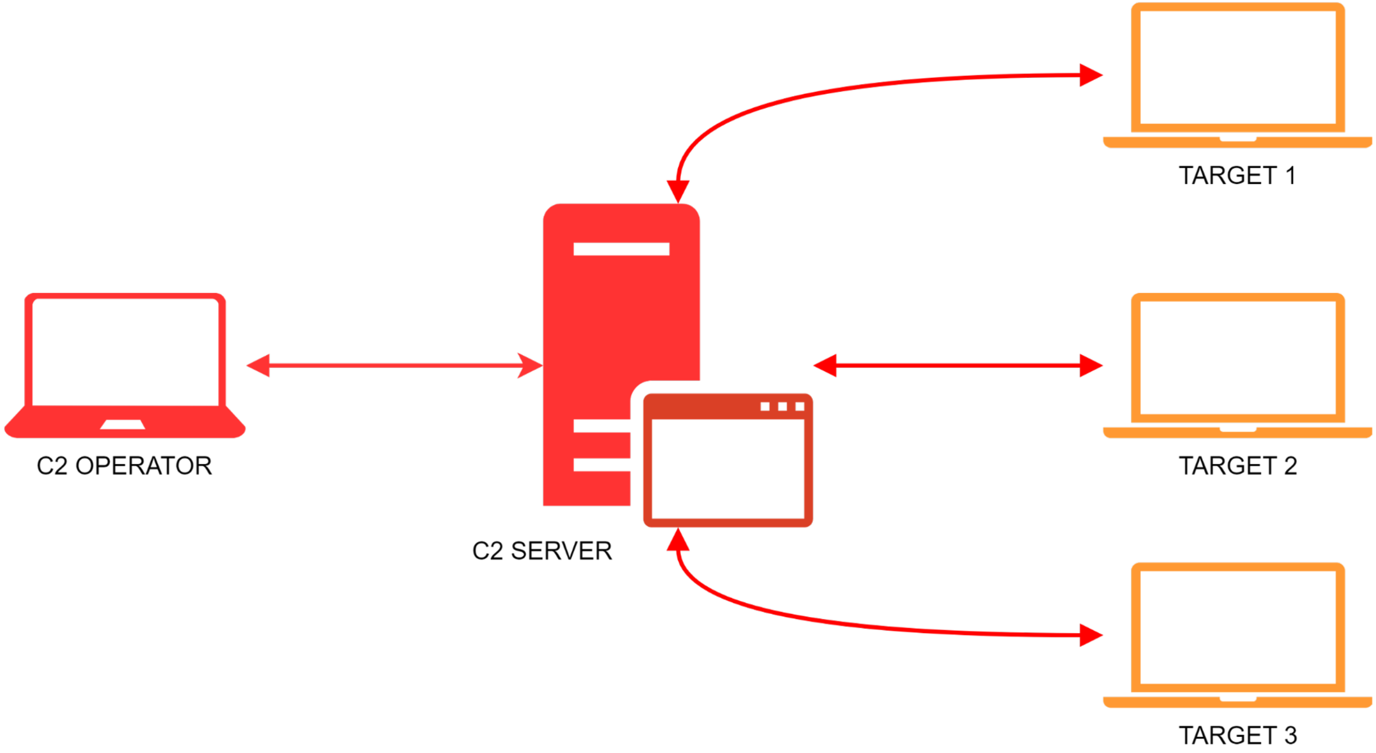
- This model utilizes the traditional client-server communication model, where the compromised host (client/agent) will call back to a centralized C2 server awaiting further instructions.
- In this model the C2 infrastructure is typically centralized in that the compromised host only communicates with a single master C2 server.
- It is important to note, however, that a centralized C2 model will typically also include the use of redirectors, proxies and load balancers to mask the IP addresses of key pieces of the infrastructure.
- Most well designed centralized C2 infrastructure will also have inbuilt kill switches designed to terminate connectivity to compromised hosts and wipe all data tied to the operation as a defense against security researchers or law enforcement.

Centralized C2 Model

- An increasingly popular trend with centralized C2 infrastructure is the use of advanced multi-channel stagers/payloads that are designed to establish communication with more than one C2 server simultaneously.

- This increases the complexity of the adversaries C2 infrastructure and makes it much more complex and time consuming to identify the primary C2 server being used to communicate with compromised hosts.

Centralized C2 Model



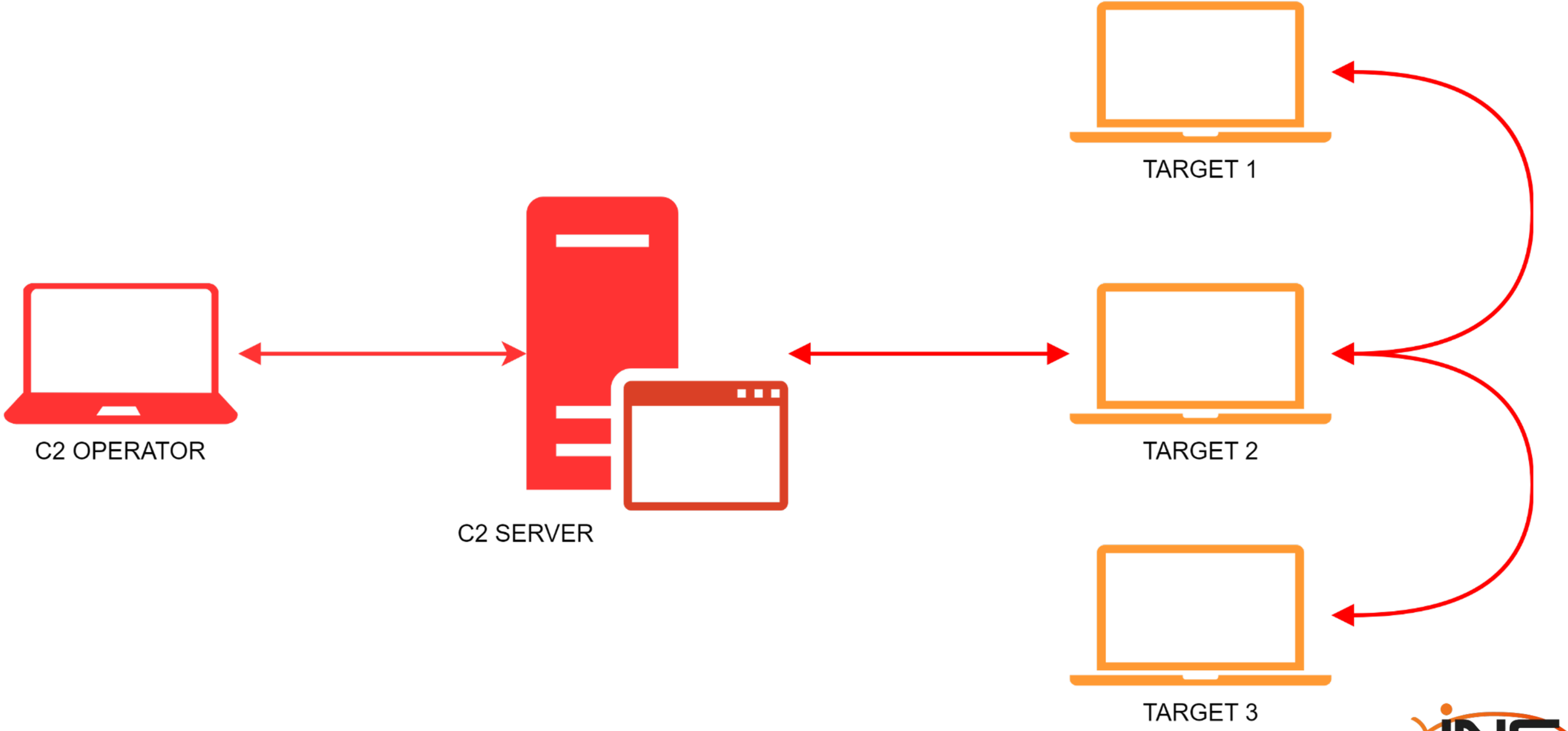
P2P C2 Model

- In a Peer-to-Peer (P2P) C2 model, communication from the C2 server is delivered through a web of botnet members that relay the commands/instructions between them.

- In this model, no single member of the botnet network is the master C2 server making it difficult to identify where the commands/instructions originated from.

- A P2P C2 model typically utilizes a single compromised host to maintain communication with the C2 server, whereby, all other compromised hosts communicate over a botnet network and transmit information through the single egress host.

P2P C2 Model



Overt/Out of Band C2 Model

- This C2 model leverages already existing communication protocols and social media platforms, IRC email and many more to facilitate their C2 infrastructure.

- The primary motivation for using this model is to take advantage of already existing platforms like Twitter or Gmail as a means of communicating with compromised hosts (implants) with the aim of evading detection.

- This C2 model can be very difficult to detect as egress traffic is in the form of communication with services like Twitter or Gmail (often marked as non-suspicious).

C2 Framework Terminology

C2 Framework Terminology

TERM	DEFINITION
C2 Server	This is the hub/server that agents call back to.
Listener	Listener process that runs on the C2 server or redirector. Listens for call backs from compromised hosts over a specific port or protocol and maintains communication between the two.
Agent	An agent is a piece of code or the mechanism that is generated by a C2 framework and calls back to a listener on a C2 Server.
Implant	Mechanism that provides interactive remote access to a target system.
Beacon/Beaconing	This refers to when a compromised host with an active implant/agent calls-back to the C2 server for instructions.
Interface	Control mechanism providing operators with interactive access to the C2 server. (Empire Client)
Payload	Piece of code executed on target system in order to achieve a specific goal like establishing a reverse shell.
Stager	A stager is a small executable that is an initial payload. It is a relatively small piece of code that is executed to prepare for a much larger and more capable payload known as the stage payload.
Sleep Timer	Sleep Timers are used to modify the rate at which an agent sends beacons to a C2 server. (Sleep timer of 10 seconds means that the agent will send a beacon out every 10 seconds)
Jitter	Jitter allows you to add some variability to the sleep timer in order to make the communication/traffic look less sequential. Some C2 Frameworks provide the ability to modify packets.

C2 Deployment & Operation

C2 Infrastructure Deployment

- When designing and deploying a C2 infrastructure, it is very important to consider the following factors:

  - Payload delivery – Method of delivering initial access payload (Email, phishing web delivery etc.).
  - Client-based protections – What sort of protection mechanisms are installed on target systems (AV, EDR, HIDS etc.).
  - Network-based protections – Network protection mechanisms present on target network (Egress filtering, IDS/IPS).

Payload Delivery

- The first step in any operation will involve obtaining initial access to a target system. This may be achieved through the use of various initial access TTPs like:
  - Exploiting Public Facing Applications
  - Supply Chain Compromise
  - Phishing
  - Valid Accounts
- Realistically speaking, initial access is rarely achieved via a C2 Framework stager.
- C2 frameworks come into the mix during post-exploitation, as a result, you must decide on the means of delivering the C2 Framework payload/stager.

Payload Delivery

- The types of stagers/payloads you can utilize will be limited to the C2 Framework you are using in addition to the type of operating system running on the target system.

- As a result, you must choose the correct C2 framework that aligns with the aforementioned factors.

- For example, it would be unwise to use a C2 Framework that utilizes Python based implants with no compilation in order to establish C2 connectivity on Windows operating systems.

Client-Based Protections

- Most C2 Framework stagers/payloads will be easily detected by a modern AV. As a result, you must also factor in evasion of client-based protection mechanisms like AVs, EDRs etc.

- The following techniques are typically used to ensure that the C2 Framework stager/payload evade signature-based AV solutions:
  - PE Encoding & Obfuscation
  - Shellcode Injection
  - PS Encoding & Obfuscation
  - PowerSploit – Shellcode injection in memory

Network-Based Protections

- Another factor to consider when designing your C2 infrastructure is the network-based protections in place on the target network.

- The following factors should be considered:
  - Egress firewall rules and filtering
  - IDS/IPS detection
  - Inline payload detection

- When designing communication channels, you must always factor in what egress traffic will look like from the target network perspective.

Network-Based Protections

- The following best-practices should be observed when deploying your C2 infrastructure:

- Ensure communication channel is over a standard port (80,8080,443).
- Note: Communication over standard ports will be heavily monitored, as a result, it is recommended to utilize domains over raw IP addresses.
- Non-standard ports like TCP 53 can be used to mask communication as legitimate DNS requests.

Domain Fronting

- Domain Fronting, is a technique utilized by adversaries in order to conceal the true destination of network traffic.
- In the case of C2 Frameworks, Domain Fronting us used to conceal the true destination of an HTTP(S) request.
- This is achieved by using a different domain name in the initial connection request rather than the actual target domain.
- This technique is further embellished through the use of encryption protocols like TLS or HTTPS in order to encrypt/obfuscate the traffic.

Domain Fronting Weaponized

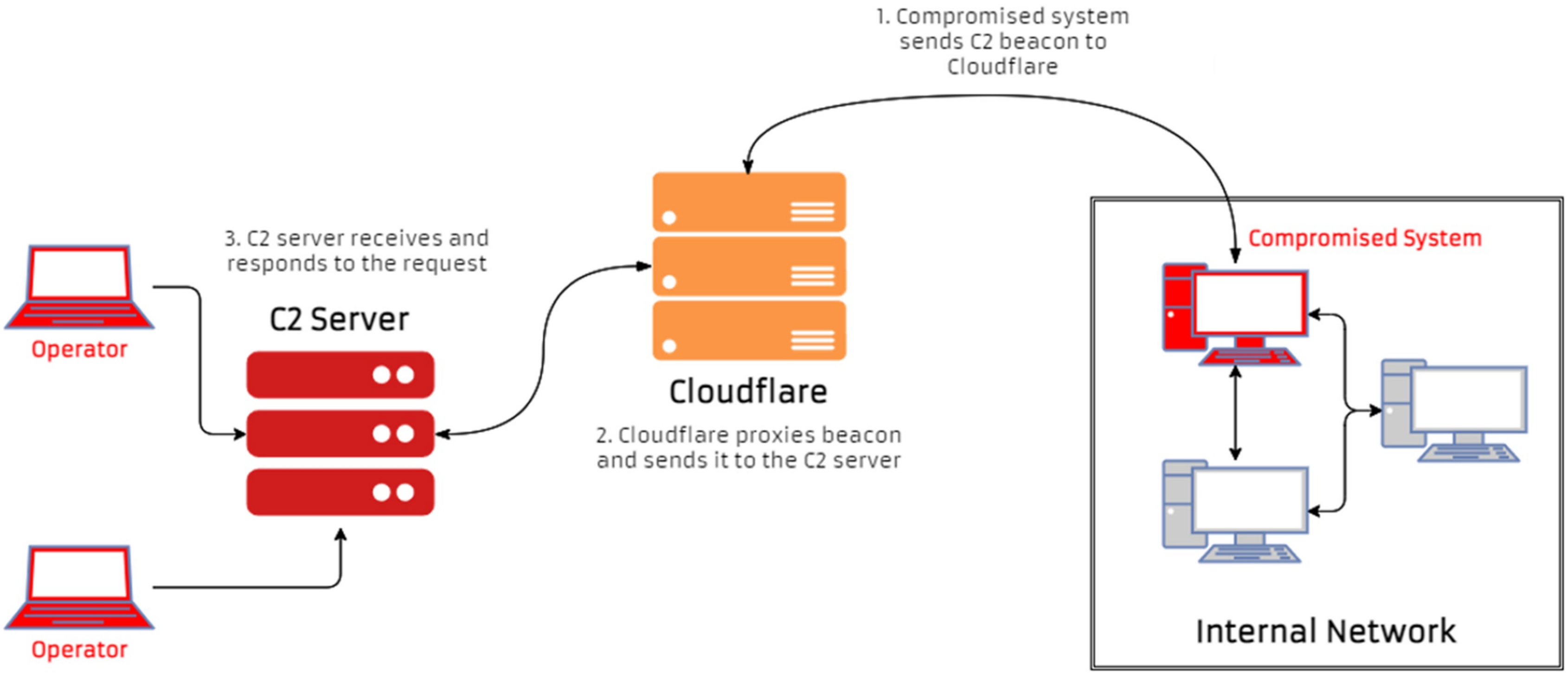
- A common service/tool that is leveraged by adversaries to facilitate Domain Fronting is Cloudflare.

- Adversaries can leverage Cloudflare to make it look like an agent is communicating with a trusted IP address (IP Block owned by Cloudflare).

Domain Fronting Weaponized

- Adversaries/Red Teamers will typically utilize the following methodology when setting up Domain Fronting with Cloudflare:
  - Adversary/Operator purchases a domain and configures it to utilize Cloudflare nameservers, consequently proxying all requests through Cloudflare.
  - The implant/payload is configured to beacon back to the C2 server domain.
  - Cloudflare receives and proxies the beacon from the agent, analyzes the host header and relays the request/response to the C2 server domain.
  - The C2 server receives the request from Cloudflare and sends a response with commands, which is also proxied through Cloudflare.
  - The response containing commands is then received by the agent via Cloudflare.

Domain Fronting Weaponized


1. Compromised system sends C2 beacon to Cloudflare
Compromised System
Operator
3. C2 server receives and responds to the request
C2 Server
Cloudflare
2. Cloudflare proxies beacon and sends it to the C2 server
Operator
Internal Network

The C2 Matrix - Choosing The Correct C2 Framework

Choosing The Correct C2 Framework

- When deploying C2 infrastructure, one question dominates the discussion. What C2 Framework should I use?

  - Over the last decade, not much was known about C2 frameworks, how they work and what differentiates them.
  - Another issue that has plagued Red Teamers has been the sheer number of frameworks available in the wild.

- This is where the C2 Matrix comes in to play.

The C2 Matrix

- The C2 Matrix was created to aggregate all the Command-and-Control frameworks publicly available (open-source and commercial) in a single resource to assist teams in testing their own controls through adversary emulations (Red Team or Purple Team Exercises).

- This allows more efficient decisions making when called upon to emulate and adversary TTPs.

The C2 Matrix

- It is the golden age of Command and Control (C2) frameworks. Learn how these C2 frameworks work and start testing against your organization to improve detective and preventive controls.

- The C2 Matrix currently has 35 command and control frameworks documented in a Google Sheet, web site, and questionnaire format.

The C2 Matrix

- Google Sheet (Golden Source): https://docs.google.com/spreadsheets/d/1b4mUxa6cDQuTV2BPC6aA-GR4zGZi0ooPYtBe4IgPsSc/edit#gid=0

- Website: https://www.thec2matrix.com/matrix

Demo: The C2 Matrix

Introduction To PowerShell-Empire

PowerShell-Empire

- Empire 4 is a pure PowerShell C2/post-exploitation framework built on cryptological-secure communications and flexible architecture.

- Empire implements the ability to run PowerShell agents without needing powershell.exe, provides you with rapidly deployable post-exploitation modules ranging from keyloggers to Mimikatz, and adaptable communications to evade network detection, all wrapped up in a usability-focused framework.

- Empire 4 includes a pure-PowerShell Windows agent, Python 3.x Linux/OS X agents, and C# agents. It is the merger of the previous PowerShell Empire and Python EmPyre projects.

PowerShell-Empire

- PowerShell-Empire was recently updated and is now officially supported and maintained by Kali Linux, more information regarding the update can be found here: https://www.kali.org/blog/empire-starkiller/

- In order to get an understanding of how Empire works and the components that make up the framework, It is recommended to go through the official wiki which can be found here: https://bc-security.gitbook.io/empire-wiki/

PowerShell-Empire

- Windows agents are purely implemented in PowerShell (without powershell.exe!), and Linux/macOS is done in Python 3.
- Feature rich with various options to bypass various protections (and allows for easy modification for custom evasion), Empire is often a favorite for Command and Control (C2) activity.
- Official GitHub Repo: https://github.com/BC-SECURITY/Empire

PowerShell-Empire Features

- Server/Client Architecture for Multiplayer Support
- Supports GUI & CLI Clients
- Fully encrypted communications
- HTTP/S, Malleable HTTP, OneDrive, Dropbox, and PHP Listeners
- Massive library (400+) of supported tools in PowerShell, C#, & Python
- Donut Integration for shellcode generation
- Modular plugin interface for custom server features
- Flexible module interface for adding new tools
- Integrated obfuscation using ConfuserEx 2 & Invoke-Obfuscation
- In-memory .NET assembly execution
- Customizable Bypasses
- JA3/S and JARM Evasion
- MITRE ATT&CK Integration
- Integrated Roslyn compiler (Thanks to Covenant)
- Docker, Kali, Ubuntu, and Debian Install Support

PowerShell-Empire Agents & Modules

Modules

- Assembly Execution
- BOF Execution
- Mimikatz
- Seatbelt
- Rubeus
- SharpSploit
- Certify
- ProcessInjection

Agents

- PowerShell
- Python 3
- C#
- IronPython 3

PowerShell-Empire Architecture

- Empire 4 introduces a new server and client architecture which requires running each in separate terminals.

- The PowerShell-Empire server is used to manage server related functionality.

- The PowerShell-Empire client is used to interact with the Empire server.

- The two primary interfaces for interaction with Empire are:
  - PowerShell-Empire Client (CLI)
  - Starkiller (GUI)

PowerShell-Empire Terminology

TERM	DEFINITION
Listener	Local process listening for a connection/beacon from a compromised host
Stager	Piece of code or script used to establish an initial foothold on a target system
Agent	Process running on the compromised host responsible for connecting to the listener
Module	Piece of code executed via the Agent to achieve a specific goal

PowerShell-Empire Listeners

LISTENER	FUNCTIONALITY
dbx	Dropbox listener (Requires token in order to interact with the Dropbox API)
http	Standard HTTP/HTTPS listener
http_com	HTTP/HTTPS listener that uses a hidden IE COM object
http_foreign	HTTP/HTTPS listener used to inject Empire payloads
http_hop	HTTP/HTTPS listener that redirects commands to another listener to conceal the initial IP address
http_mapi	HTTP/HTTPS listener that uses the Liniaal utility allowing you to gain control over the target host through an Exchange server

PowerShell-Empire Stagers

LISTENER	FUNCTIONALITY
bash	Standard bash script
launcher	One-liner written in a specific scripting language (vbs, bat etc.)
macro	Macro for the Office Suite
jar	JAR payload
shellcode	Windows shellcode
Csharp_exe	PowerShell C# PE
dll	DLL Stager
hta	HTA stager for IE/mshta.exe

Demo: PowerShell-Empire

Red Team Ops With PowerShell-Empire

Demo: Red Team Ops With PowerShell-Empire

Red Team Ops With Starkiller

Starkiller

- BC Security, the company responsible for maintaining PowerShell-Empire also developed a GUI interface companion for Empire called Starkiller.

- Starkiller is a Frontend for Powershell-Empire. It is an Electron application written in VueJS and provides users with an intuitive way of interacting with Empire.

Starkiller Features

- Interactive agent shell.
- Malleable profile management.
- Ability to enable/disable modules.
- Process browser.
- File browser.
- Chat widget for collaborative ops.

Lab Demo: Red Team Ops With Starkiller

Command and Control (C&C/C2)

Course Conclusion

Learning Objectives:

- You will have an understanding of what Command and Control is with regards to Red Team/Offensive operations.
- You will have an understanding of what C2 Frameworks are, how they work, what functionality they offer and the role they play in red team operations.
- You will have an understanding of the various communication models and protocols used in designing, deploying and operating C2 infrastructure .
- You will be able to identify the correct C2 Framework to use based on the nature of engagement you are performing in addition to the features you require for a successful operation.
- You will have the ability to install, configure and effectively use PowerShell-Empire and Stakiller for Red Team operations in Windows environments.
- You will have the knowledge and experience in using some of the most popular C2 Frameworks available.

Thank You!

EXPERTS AT MAKING YOU AN EXPERT

INE